Oil & Gas Science and Technology – Rev. IFP, Vol. 63 (2008), No. 6, pp. 723-736
Copyright © 2008, Institut français du pétrole
DOI: 10.2516/ogst: 2008042

A New Modelica Model and Scicos Simulation for 0D/1D Nonlinear Complex Systems

M. Najafi[1] and Z. Benjelloun-Dabaghi[2]

[1] INRIA, Domaine de Voluceau – Rocquencourt, BP 105, 78153 Le Chesnay Cedex - France
[2] Institut français du pétrole, IFP, 1-4 avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex - France
e-mail: masoud.najafi@inria.fr - zakia.benjelloun-dabaghi@ifp.fr

Résumé — Une nouvelle approche de modélisation et simulation de systèmes 0D/1D complexes non linéaires à l'aide de Modelica et Scicos — Le but de cet article est de montrer que, grâce à des méthodes et outils numériques adaptés, la résolution, le contrôle et la compréhension des phénomènes physiques complexes sont aujourd'hui possibles. Dans ce travail, après une brève description du langage Modelica et de la méthode numérique "SUNDIALS", utilisée par le simulateur Scicos pour résoudre les systèmes non linéaires 0D/1D, nous traitons deux exemples industriels : en forage, pour résoudre un large système 1D complexe implicite, puis un exemple en contrôle moteur, afin de tester les capacités de ce langage à résoudre dans un même bloc d'équations, la commutation de modèles correspondant à un système 0D hybride (cohabitation du temps continu et discret). Ces modèles ont déjà été mis au point à l'IFP, avec la boîte à outils Simulink du logiciel Matlab. Pour le cas du forage, le modèle Simulink est construit avec 116 sous-systèmes et plus de 500 blocs, alors que le modèle Scicos n'est construit qu'avec 9 Scicos/Modelica blocs. Le modèle moteur Simulink est composé de 203 blocs et 30 sous-systèmes, alors que le modèle en Scicos/Modelica est composé de moins de 20 blocs. Le temps de simulation à l'aide de Simulink varie entre 6 et 150 secondes selon le type de l'intégrateur. En Scicos/Modelica, avec l'intégrateur SUNDIALS, la simulation prend moins de 3 secondes.

Abstract — A New Modelica Model and Scicos Simulation for 0D/1D Nonlinear Complex Systems — *The purpose of this article is to show that an improvement in understanding of physical phenomena can be achieved today, with the help of suitable numerical methods and simulation tools. Modeling and simulation are becoming more crucial, since engineers need to analyze very complex systems composed of several components from different domains. These systems are generally simulated by solving differential-algebraic systems which are in general hybrid systems incorporating many variables. The Modelica language allows formal writing of mathematical equations and the expression of models in an acausal way. In this paper, first, we present the advantage of the Modelica language in modeling and simulation of industrial applications. Then, two industrial test cases will be explained: the drilling-well station, which is a large complex 1D implicit system, and an automotive application where we test the advantages of Modelica to simulate switched models, known as commutated models. These applications have already been developed at IFP by Simulink (a Mathworks product). In the drilling case, the Simulink model was built with 116 subsystems and more than 500 blocks, whereas the Scicos model was built with just 9 Scicos/Modelica blocks. The engine model in Simulink is composed of 203 blocks and 30 subsystems, whereas the model in Scicos/Modelica is composed of less than 20 blocks. The simulation time in Simulink varies between 6 and 150 seconds as a function of the selected numerical solver, while in Scicos/Modelica it takes less than 3 seconds.*

INTRODUCTION

Scilab (*www.scilab.org*) is free and open-source software for scientific computing, and Scicos (*www.Scicos.org*) is a toolbox of Scilab and provides an environment for modeling and simulating dynamical systems [4]. The underlying formalism in Scicos allows modeling of very general dynamical systems: systems including continuous, discrete-time and event-based behaviors, which are generally referred to as hybrid systems.

Modeling, in general, can be classified into two major categories: causal and acausal modeling. In causal modeling, the model is decomposed into several modules. Each module may have several inputs and outputs. The evaluation of outputs is based on the inputs and the internal variables. Acausal modeling, on the other hand, is closer to modeling with physical components. In a physical system, it is not possible to classify (at least a priori) a quantity or a variable as input or output. The causality of a variable depends on the complete set of the model's equations.

Scicos is originally based on causal system modeling. In order to implement a causal module, explicit blocks with explicit inputs and outputs (standard Scicos blocks) can be used. Recently, an extension of Scicos has been developed to allow acausal modeling or modeling of physical systems with components. This has been done, in particular, by lifting the causality constraint on Scicos blocks and by introducing the possibility of describing block behaviors with mathematical equations [11].

Introducing acausal modules into Scicos required the use of a new block type, *i.e.*, *implicit blocks* or *components*. The advantage of implicit blocks is providing facilities to model a physical system by interconnecting components. In order to use components in addition to standard Scicos blocks, several new features have been added to Scicos. Components or implicit blocks are interfaced via special links associated with physical quantities such as current or voltage in electronics, or flow or pressure in hydraulics. The connection between two components represents a real physical connection.

Internal and input/output behavior of standard explicit blocks is written in C or in Fortran, which are imperative or sequential languages. In order to describe the behavior of implicit blocks the Modelica language was used. In other words, the mathematical formulas describing the behavior of components are written in the Modelica language.

Modelica (*www.modelica.org*) is a modern object-oriented programing language based on equations instead of assignment statements. It has a multi-domain modeling capability. For example, electrical, mechanical, thermal hydraulic, hydraulic and control systems can all be described in Modelica. In Modelica, equations are composed of expressions both on the left hand side and the right hand side. It is not required to write the equations in the form of assignments, nor to write the equations in a specified order.

The Modelica compiler of Scicos performs symbolic manipulations to simplify the model and generate an index-1 DAE. Modelica language is independent of the simulation tool and has been developed by the Modelica association [8, 10, 11].

In Modelica, in order to write a complicated model easily and efficiently, the model is decomposed into several other components. Each component, which can also be composed of other subcomponents, is modeled in Modelica. Then, by interconnecting the components the whole model can be constructed.

This paper is composed of two parts. In the first part, the model of a spark-ignited engine, the engine components, and the whole engine model will be presented. In the second part of the paper, the model of a drilling-well station and its components, as well as the whole model, will be explained.

1 MEAN VALUE SI ENGINE

The model of the Spark-Ignition (SI) engine described in this paper is a nonlinear, low-frequency model of a fuel-injected four-cylinder SI engine, which is generally referred to as a *mean value* model. Mean value engine models attempt to capture dynamics in a time-scale spanning over several combustion cycles. Fast events are not of interest other than their effects on a larger scale. Most cyclic dynamics are modeled by their average value over a cycle. The speed and torque output of the engine and the pressure in the inlet manifold are the aspects of most interest in the mean value engine model that we have developed. The mean value engine model generally represents a basis for the development of different engine control strategies.

The model of the overall engine is composed of several subsystems or components. In order to develop the model of the SI engine more easily, the engine subsystems are modeled separately. The elements of the system including the air throttle, the intake manifold, Exhaust Gas Recirculation (EGR), the canister purge mechanism, sensor dynamics, the combustion chamber, and the load and speed perturbations are modeled. Inherent system delays in the four-stroke engine cycle including the induction-to-power stroke delay, effects of the air/fuel ratio or fuel richness are not modeled in this paper. The system including fundamental components, sensors and actuators is illustrated in Figure 1.

In the following subsections, a brief description of the engine components corresponding to the physical phenomena used in our model will be given [1, 6, 7, 9, 10, 14].

1.1 Air Intake Throttle

The air throttle controls the air flow rate into the air manifold and the combustion chamber. The air throttle can be modeled as a flow restriction. The model of a flow restriction highly

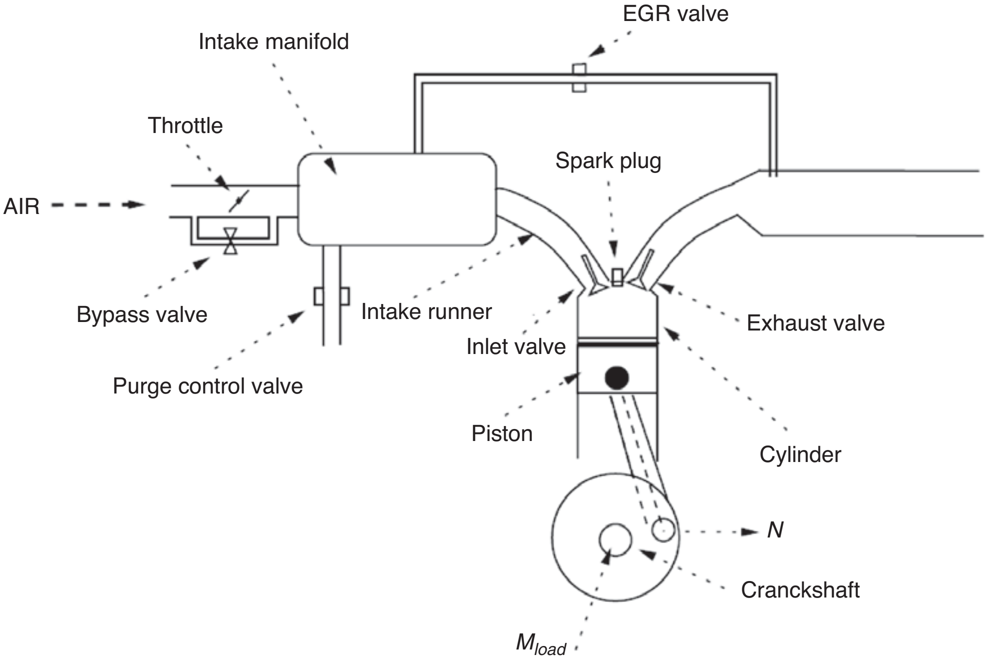

Figure 1

Principle sketch of SI engine.

depends on the pressure difference across the restriction: if small enough, the gas density is considered equal on both sides, *i.e*., the gas is considered as an incompressible fluid. If, on the other hand, large pressure differences can be expected the restriction should be modeled assuming compressible fluids. Thus, the intake air mass flow rate (M_{air}) is modeled with equations.

$$\dot{M}_{air} = (\alpha + k\alpha_0)\frac{P_{air}}{\sqrt{R_{air}T_{air}}}\psi(\frac{P_{man}}{P_{air}})^{0.5} \tag{1}$$

where α and α_0 are the throttle plate open area and bypass air passage, respectively. P_{man}, P_{air}, R_{air} and T_{at} are the manifold pressure, the environmental air pressure, the air specific gas constant and the environmental air temperature, respectively. Ψ (x) is a scalar function defined as:

$$\Psi(x) = \begin{cases} \frac{2\gamma}{\gamma-1}\left(x^{\frac{2}{\gamma}} - x^{\frac{\gamma+1}{\gamma}}\right) & \text{if } x \geq \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \\ \gamma\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}} & \text{otherwise} \end{cases} \tag{2}$$

where $\gamma = C_p/C_v$ is the ratio of specific heats of the air. Note that we have also assumed that there is no back flow and the temperature is unchanged across the throttle. Interested readers are referred to [8] for more details.

In the throttle, α is controlled by the accelerator pedal. When the engine is turning in the idle mode, *i.e*., $\alpha = 0$, the necessary air for maintaining the minimum power of the engine is supplied through an air passage, called an air bypass passage. The bypass area α_0 is controlled by the engine control unit (ECU) to maintain the necessary air flow during idle speed control.

A schematic model of the throttle block in Scicos is given in Figure 2. The air throttle component has two implicit ports and two explicit inputs. In Figure 2, the square ports are implicit and triangle ones are explicit. Implicit ports represent inlet and outlet air flows and explicit input ports represent control signals for α_0 (the bypass opening area and α the throttle plate). The implicit ports, modeled with the *connector* clause in Modelica, carry three variables, *i.e*., pressure (P),

mass flow rate ($\dot{M}$) and temperature (T). The Modelica code for the air connector follows:

```
connector PortAir
  Real P   "Air pressure";
  Real phi "Air mass flow rate";
  Real T   "Air temperature";
end PortAir.
```

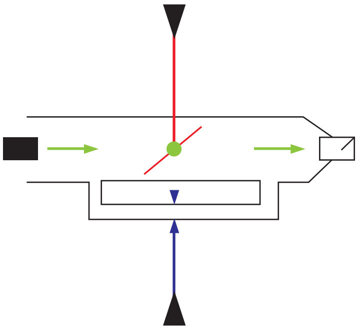

Figure 2

A schematic diagram of the air inlet throttle block in Scicos.

1.2 Exhaust Gas Recirculation (EGR)

In order to reduce harmful emissions resulting from combustion, some of the exhaust gas is diverted back into the combustion process. In this method, the inlet and exhaust manifolds are connected with a pipe and the recirculated gas flow rate is controlled by a valve. In order to reduce the formation of nitrogen oxides, the EGR system often includes a cooler to reduce the temperature. That is why the thermal effects of the EGR are not modeled in this paper. The rate of the gas recirculation is typically less than 25 percent of the total amount of exhaust gases in SI engines and up to 50 percent in diesels [9]. The first approach to modeling the total gas mass flow through the EGR control valve, ignoring the back flow, is using the same formula used for the throttle model, *i.e.*:

$$\dot{M}_{egr} = \beta \frac{P_{egr}}{\sqrt{R_{egr} T_{egr}}} \psi(\frac{P_{man}}{P_{egr}})^{0.5} \tag{3}$$

where β is the open valve area as a function of the PWM control signal coming from the ECU. P_{egr} is the pressure just before the exhaust control valve. Note that P_{egr} must be measured, but for the sake of simplicity, we assume that it is equal to the atmospheric pressure. The specific gas constant R_{egr} and γ depend on the combustion process, but we assume that they are equal to those of the air.

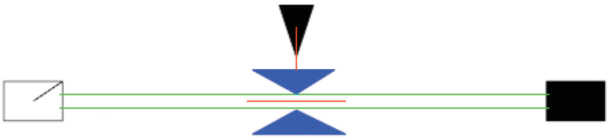

Figure 3

A schematic diagram of the EGR block in Scicos.

A schematic model of the EGR block in Scicos is given in Figure 3. The EGR block has two implicit ports and one explicit input port as the signal for the EGR control valve.

1.3 Canister

Most of the HC emissions in modern cars are from the exhaust, but a considerable part also comes from evaporative losses in the fuel tank. In older cars, fuel vapors in the tank are released directly to the atmosphere through a vent in the fuel-tank filler-cap. Most modern cars use an evaporative emissions management system to reduce these emissions. The basic function of this system is to trap and store the fuel vapors from the fuel tank in a canister until the engine is started. Thereafter, the trapped fuel vapors are drawn into the engine by air intake manifold and combusted. To control the flow of vapors into the engine, a purge control valve with no back flow is used. Similar to the model of the EGR system, the mass flow rate of the air/fuel mixture through the canister purge is modeled as a restriction, *i.e.*:

$$\dot{M}_{can} = \xi \frac{P_{can}}{\sqrt{R_{can} T_{can}}} \psi(\frac{P_{man}}{P_{can}})^{0.5} \tag{4}$$

where ξ is the open valve area as a function of the PWM control signal coming from the ECU, and P_{can} is the pressure before the purge control valve. Note that P_{can} must be measured, but for the sake of simplicity, we assume that it is equal to the atmospheric pressure. The specific gas constant R_{can} and γ depend on the fuel concentration of the vapors, but the overall dependency of the concentration in Equation 4 can be shown to be so small that it might be neglected [6]. A schematic model of the canister in Scicos is given in Figure 4. This block is modeled with only one implicit port that is connected to the air manifold, and an explicit input signal for the control valve also carries the fuel concentration in the air. The implicit port of the canister carries four variables: pressure P, mass flowrate $\dot{M}$, temperature T, and the fuel concentration in the air.

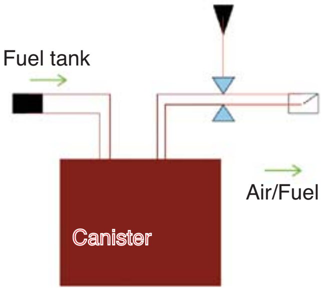

Figure 4

A schematic diagram of the canister block in Scicos.

1.4 Intake Manifold

The air passing through the air throttle, the EGR and the canister are mixed in the intake manifold and are sent into the combustion chamber through the intake runner. In SI engines the inlet manifold pressure is reduced by a throttle in order to control the output torque. The intake manifold is seen as a volume and the throttle, purge control valve and intake runners are considered as restrictions. We have assumed an isothermal manifold heat transfer, *i.e.*, constant manifold air temperature.

The air in the intake manifold is composed of fresh air, fuel and burnt gas. Assuming that the concentration of fresh air Y_{air} is much bigger than that of the burnt gas Y_{ex} and fuel Y_f, *i.e.*, $\{Y_{ex}, Y_f\} << Y_{air}$, we can define the relative mass concentrations in the intake manifold. The concentrations can be described as functions of the partial pressures of fuel and air in the intake manifold, *i.e*:

$$\begin{aligned} M_{man}\dot{Y}_{ex} &= \dot{M}_{egr} - Y_e \dot{M}_{run} \\ M_{can}\dot{Y}_f &= \dot{M}_{can} - Y_f \dot{M}_{run} \\ Y_{air} &= 1 - Y_e - Y_f \end{aligned} \tag{5}$$

where M_{an} and $\dot{M}_{run}$ are the total mass of gases in the intake manifold and mass flow rate through the intake runner, respectively. Using the ideal gas law, we can obtain the following model for the intake manifold pressure.

$$\begin{aligned} \dot{P}_{man} V_{man} &= R_{air}N_{air}(\dot{M}_{air} T_{air} + \dot{M}_{egr} T_{egr} \\ &+ \dot{M}_{can} T_{can} - \dot{M}_{run} T_{man}) \\ \dot{M}_{man} &= \dot{M}_{air} + \dot{M}_{egr} + \dot{M}_{can} - \dot{M}_{run} \\ M_{man}R_{air}T_{man} &= P_{man}V_{man} \end{aligned} \tag{6}$$

As will be seen later on, the flow rate in the intake runner $\dot{M}_{run}$ is imposed by the pumping mechanism of the combustion chamber and the crankshaft rotation.

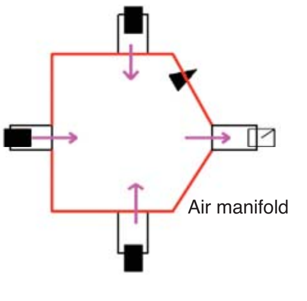

Figure 5

A schematic diagram of the manifold block in Scicos.

A schematic model of the manifold diagram in Scicos is given in Figure 5. The block has four implicit ports and one explicit output port representing the MAP (Manifold Air Pressure) sensor.

1.5 Combustion Chamber

The combustion chamber is the heart of the engine. The air/fuel mixture flows into the cylinders and reacts, and usable energy is extracted from the heated gas which is then expelled. In an SI engine, the air is mixed with fuel before entering the cylinders. In this paper, the effects of the air/fuel ratio are not modeled. The cylinder is continuously swept by a piston which is connected to the crankshaft via a rod. The top of the cylinder houses intake and exhaust ports, and a spark-plug in SI engines. The SI engine works in four discrete strokes. In the first stroke, fresh air is inhaled through the intake ports as the piston moves down. In the second stroke all valves are closed while the piston moves up and compresses the air. At the peak of compression, fuel is injected and the expansion stroke starts. In the fourth stroke the exhausts are ejected through the exhaust ports.

The cylinder and the crankshaft have two important roles: torque generation and air pumping. During the third stroke, when gas burns and expands, the piston is forced down. The downward movement is then transformed into rotational movement. The applied torque on the crankshaft depends on several factors, such as the air/fuel mixture ratio, spark ignition time, manifold pressure, angular velocity of the crankshaft, etc. Since there is no accurate and simple physical model describing the generated torque, it is quite often that a map is used. This map gives the optimum generated torque as a function of the manifold pressure and the angular velocity of the crankshaft ω. Thus, we have $\tau^{opt}_{gen} = f(\omega, P_{man})$ where τ^{opt}_{gen} is the optimum deliverable torque by the power stroke. This map gives the value of the produced torque regardless of other important effects such as the effects of spark advance.

Adjusting the spark advance timing, we can optimize engine efficiency to deliver peak combustion pressure when the piston reaches about 10° after the top dead center (TDC) angle. Incorrect spark timing can have a significant effect on emission output and vehicle drivability. The amount of the spark advance needed by the engine varies as a function of the number of different operating conditions. Generally, spark advance is adjusted based on the following strategy: spark advance increases with higher engine speeds for performance and fuel economy; spark advance needs to decrease under heavy load conditions to avoid detonation. When there is too much spark advance, particularly during high engine load conditions, the likelihood of engine detonation (knock) increases and the combustion temperature and pressure increase. This results in an increase in HC and NOx output, decreased engine performance, and possible permanent damage to the engine. Too little spark advance causes only partial combustion of the air/fuel charge, resulting in very poor engine performance and fuel economy. Partial combustion will also result in an increase in CO levels. Several variables of the system should be considered when determining the proper spark lead time. The coolant temperature, fuel quality and engine load are just a few of the many factors that can significantly impact the ideal ignition time [1]. The effects of the spark timing on the produced torque are obtained by using an experimentally obtained map. The map that we used in our simulation gives the spark advance efficiency or the ratio of the produced torque with respect to the optimal torque, *i.e.*:

$$\eta = \frac{\tau_{gen}}{\tau_{gen}^{opt}} = H(|SA|)$$

The up/down movement of the cylinder creates a pumping effect; when the piston moves downward, the air is inhaled from the intake manifold and when the piston moves upward, the burnt air is exhaled to the exhaust manifold. In an internal combustion engine the pressure on the intake side will normally be lower than that of the exhaust side. Pumping gas from low to high pressure costs energy and this energy is taken from the crankshaft. The effect is largest in SI engines on low load. The amount of the pumped air depends on several parameters such as the cylinder volume, the angular velocity of the crankshaft, pressure in the intake manifold, pressure in the exhaust manifold, and the air temperature. Since there is no accurate and simple physical model describing the gas flow rate through the intake runners, an experimentally obtained map is usually used. This map describes the total gas flow rate as a function of manifold pressure P_{man} and engine speed ω, *i.e.*, $G(\omega, P_{man})$. It is well known that we can write the air mass flow rate in the runner with:

$$\dot{M}_{run} = G(\omega, P_{man}) \frac{\omega P_{eman} V_{cyl}}{T_{manx} R_{air}} \frac{1}{120}$$

The maps used in our model were obtained at IFP for a four-cylinder SI engine. A schematic diagram of the combustion chamber in Scicos is given in Figure 6. This block has two implicit ports for the air intake runner and the exhaust outlet and another implicit port for the connection with the crankshaft. The block has one explicit input port for the spark advance signal coming from the controller.

1.6 Crankshaft Dynamics and Perturbations

The crankshaft is modeled using Newton's second law for rotating masses. This results in the following equations:

$$\begin{aligned} J\dot{\omega} &= \tau_{in} - \tau_l - \tau_{fric} \\ \tau_{in} &= (1+\sigma_1)\tau_{gen} \\ \tau_l &= \tau_{load} + \sigma_2 \\ \tau_{fric} &= k(\omega + \sigma_3) \end{aligned} \qquad (7)$$

where τ_{in} is the total produced torque that includes all perturbations due to instabilities in combustions, differences in generated torque in cylinders, and variations in fuel injection in different cylinders. The term σ_1 is a zero mean random noise. τ_l represents the load applied on the engine including controllable loads such as effects of A/C or antifrost systems on the engine, and uncontrollable perturbations modeled with a zero mean random noise. The term τ_{fric} represents the friction losses. The variation in the crankshaft speed is also modeled with a zero mean random noise.

A schematic diagram of a load block in Scicos is given in Figure 7. This block is connected to the combustion chamber block via an implicit port representing the mechanical connection of the crankshaft to the combustion chamber. The block has an explicit output port which provides the rotation speed ω.

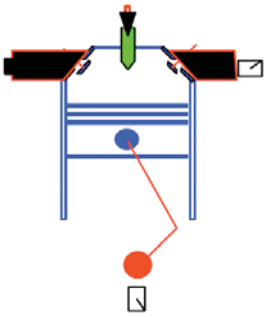

Figure 6

A schematic diagram of the combustion chamber block in Scicos.

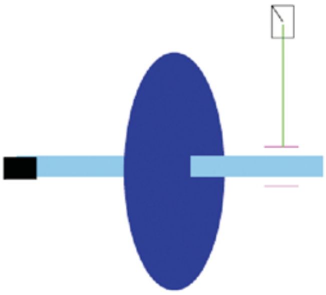

Figure 7

A schematic diagram of the load block in Scicos

1.7 Intake Manifold Air Pressure (MAP) Sensor

For all measurement systems, one desires that a sensor is fast compared with the system it monitors. If the sensor is fast, its dynamics can usually be neglected when modeling the system. In the case of the MAP sensor used in cars, this is not the case. First, the MAP sensor is usually mounted some decimeters away from the intake runner, giving a pure time delay in the measurement which depends on the speed of the intake gases, which secondly depends on the engine speed ω. Second, the sensor response is somewhat slower than the dynamics of the gases in the exhaust manifold [7]. As it is, the MAP sensor has to be modeled too. A first-order model of the sensor can be described as:

$$k_{sensor}\hat{P}'_{man} = P_{man} - \hat{P}_{man}$$

where $\hat{P}_{man}$ is the measured value for P_{man}. k_{sensor} is a time constant that should be identified.

1.8 The Model of the SI Engine

In this section, the Modelica blocks developed in the previous section are interconnected to build up the model of an SI engine. The engine model is then used to validate start-up and idle speed control strategies on the engine. In these control strategies, the objective is to do an engine cold-start and regulate the idle engine speed as low as possible in the presence of several perturbation sources. The model of the engine is given in Figure 8. In this figure, the engine is modeled with the Modelica components while the controller is developed with standard explicit Scicos blocks. Although it is possible to model the controller in Modelica, its modeling with explicit blocks in Scicos has the advantage of using the rich control toolbox of Scilab for controller design. For example, we can design PID and LQG controllers with Scilab and test their efficiency directly in Scicos.

In this simulation, the selected controller is relatively simple, *i.e.*, a PI controller. The idle engine speed controller will be active as soon as the engine speed exceeds 700 RPM. During the start-up, the spark advance is set to 20° and α_0 = 15%. When engine speed supersedes the 700 RPM threshold, the control is handed over to the PI controller that adjusts the spark advance and the bypass area as a function of the reference speed, *i.e.*, 750 RPM and an instantaneous MAP sensor and the engine speed. The results of a start-up from zero RPM and an idle speed control simulation are given in Figures 9 and 10. In this simulation, different loads

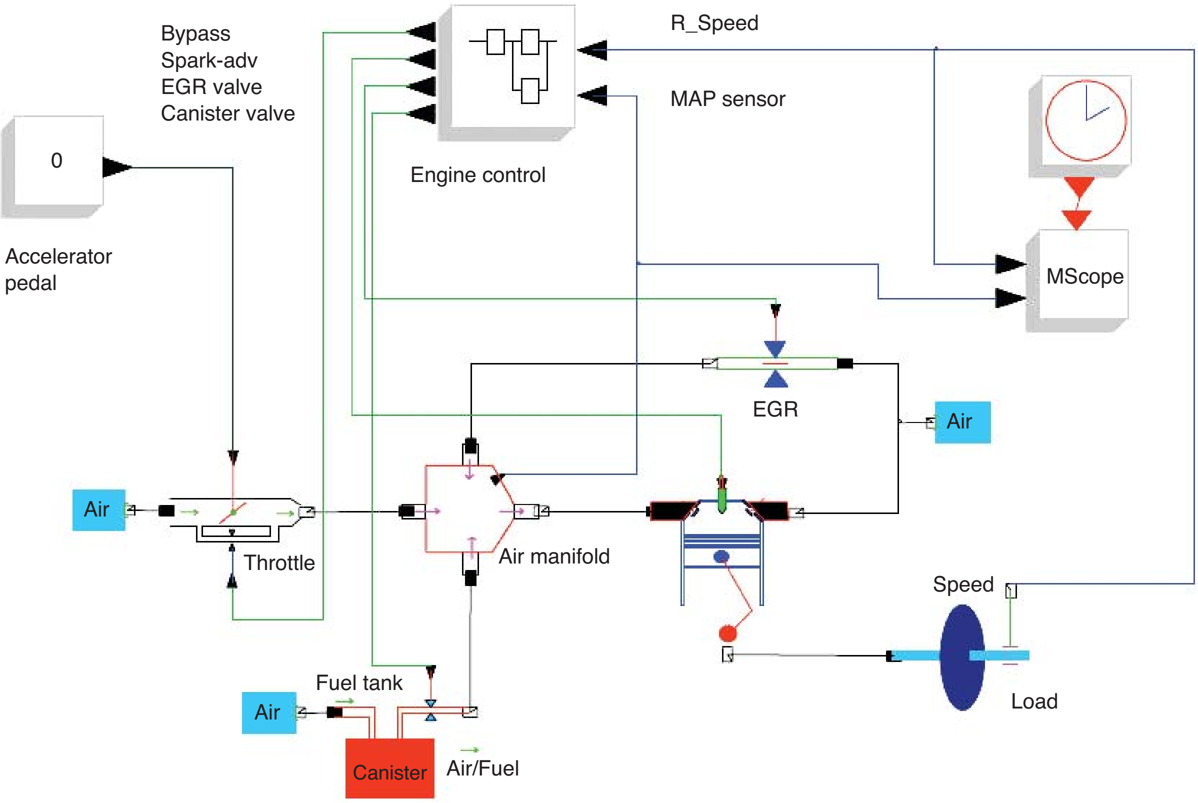

Figure 8

The Scicos model for a mean SI engine.

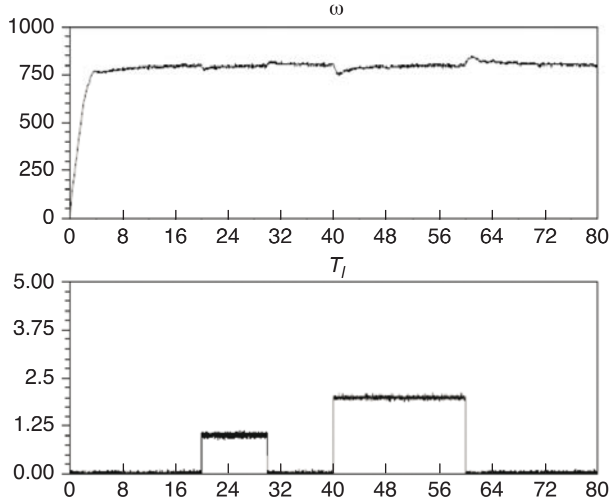

Figure 9

Top: angular velocity of the crankshaft ω; Bottom: applied load τ_l.

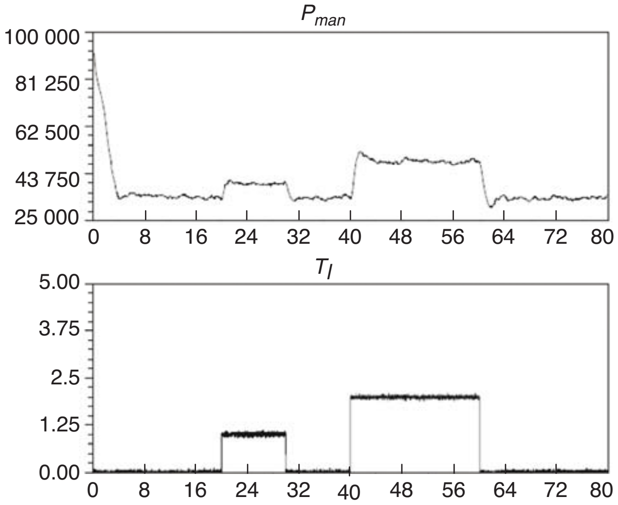

Figure 10

Top: intake manifold pressure P_{man}; Bottom: applied load τ_l.

are given at instants of 20 and 40 seconds. In the bottom plot of the figures the load are given. In the top plot of Figure 9 the engine speed is shown to be relatively regulated around 750 RPM in spite of the loads and random perturbations. The graph in the top plot of Figure 10 gives the intake manifold pressure which reduces from atmospheric pressure as the engine starts up and varies as the load changes.

1.9 Comparison

The model SI engine described in this paper has already been modeled and simulated in Simulink at IFP [2]. The modeling in Modelica was done to compare two modeling methods as well as their simulation performances. The fact that the SI engine has been modeled in Modelica has drastically facilitated the design and debugging of the model. The Simulink model is composed of 203 blocks and 30 subsystems while the Modelica model, as seen in Figure 8, is composed of nine Modelica blocks.

In order to compare the simulation performances, since the Modelica compiler generates a C code for the model, the comparison between two simulators should be done with the accelerator option of Simulink selected. Without this option, depending on the solver choice, the simulation time varies between 6.5 and 150 seconds. While with the accelerator option, the simulation time is almost the same as that of Scicos, *i.e.*, 3.8 seconds. Another great advantage of using Modelica lies in modeling systems described by DAEs. The Modelica compiler can generate code for DAE models and Scicos uses the SUNDIALS solver, which can handle fully implicit DAEs, to integrate DAES and ODES. In contrast, Simulink can only integrate semi-explicit index one DAEs. Reformulation of fully implicit DAEs is needed before they can be solved in Simulink.

2 DRILLING-WELL MODEL

Drilling a well into a reservoir is an expensive, risky, and time-consuming process. So the problems and malfunctions should be detected as soon as they appear. Most of problems in drilling industry are due to lack of a complete knowledge about the environment and the process. Modeling and simulation are inevitable to detect and control of such problems. Thus, the drilling process is monitored at the well head to detect malfunctions such as torque fluctuations produced by Coulomb frictions and torsion, and longitudinal and lateral motions. In order to suppress or control this phenomenon, a model of the well assembly is necessary.

The drilling-assembly model is used to elaborate estimators or even controllers. We used Modelica language and formal computing to model and simulate the drilling model. First, different components of a drilling well are detailed. Then, a mathematical model for main components of the drilling system will be given. At the end, a simulation of the drilling process showing torque fluctuations will be presented**.**

The drilling model that will be explained in this paper is a set of differential equations describing different components of the drilling rig, including the bit and the rock interactions. The model should be as simple as possible to explain the desired malfunctions. In our model, we will not consider the swivel and effects of drilling fluids, or the whirling in the drill string.

The diagram in Figure 11 shows the model composed of four main components: a rig, a drill pipe, a drill collar and a drilling bit. These components interact with each other via four main variables:

- *T*: the torque that a component applies on another;
- *F*: the force that a component applies on another;
- Ω: the angular velocity of a component;
- *V*: the longitudinal velocity of a component.

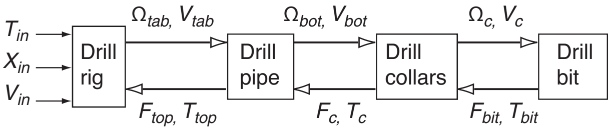

Figure 11

Connecting variables.

Note that we have assumed that only the angular and longitudinal velocity of the kelly can be controlled and measured. In the following sections, a detailed model of each component will be given.

2.1 Drilling-Well Components

A drilling-well station is composed of several parts. The first visible part of a drilling well is the rig. A drilling rig is a structure housing equipment used to drill into underground reservoirs for water, oil, etc. The basic components of a rotary drilling rig are the *derrick* and hoist, swivel, kelly, turntable, drill pipe, bit, and pump, as shown in Figure 12.

The *derrick* is the support structure that holds the drilling apparatus and the drill string.

The *drill string* consists of the swivel, *kelly*, drill pipe, drill collars and bit.

The *swivel* is a large handle that holds the weight of the drill string and allows the drill pipe to rotate while connected to the drill string.

The *swivel* is connected to the *kelly*.

The *kelly* is a four- or six-sided pipe that transfers rotary motion of the rotary table to the drill string.

The *rotary table* is a circular table in the derrick floor which is rotated by the electrical or diesel engines.

The *kelly* passes up through the rotary table and transfers the rotary motion to the drill string.

When rotated by the rotary table, the swivel and kelly are free to be raised or lowered by a cable that runs from the swivel over the sheave at the top of the derrick down the kelly drum. The kelly drum brakes and wire rope control the pressure and rate of feed to the bit by holding off or applying the weight of the drill pipe.

The *drill pipe* is always the longest component in a drill string. Typically thousands of meters of drill pipe are used to drill an oil well. The drill pipe is manufactured in segments of 10-meter lengths. Each 10-meter segment, male on one end and female on the other, is referred to as a joint. The segments are joined together as the depth of the well increases. The top joint of the drill pipe is connected to the kelly. After drilling 10 meters, the kelly must be raised and another joint of pipe added below the kelly. This is called making a connection. The bottom joint of the drill pipe is directly connected to larger-diameter pipes called *drill collars*. One of the earliest problems drillers encountered in rotary drilling was that of keeping their boreholes straight. The deeper drillers went, the more the boreholes deviated from vertical. Drill collars weigh more than the drill pipe and are designed to lower the center of gravity of the drill pipe. This helps control drilling, *e.g.*, making a straight hole, and prevents the pipe from kinking and breaking. Two to twenty drill collars are often used.

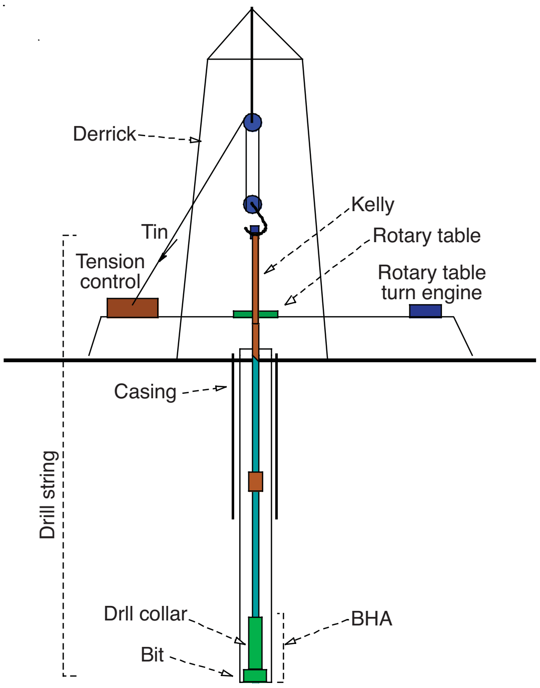

Figure 12

A schematic diagram of a drilling-well station.

The *drilling bit* is the end of the drill string that actually cuts up the rock. The drilling bit comes in many shapes and materials specialized for various drilling tasks and rock formations. The bit screws into the bottom of the drill collars. The most common bit is the tricone bit which has three rotating cones. The cones have teeth that are designed to chip and flake away the rock as the bit is rotated.

The rotating bit becomes very hot: in order to reduce the temperature of the bit, it is necessary to clean the cutting tool properly, which must rise in the hole fast enough to carry the cuttings with it, and maintaining the pressure in the hole to preserve the formation, a fluid (drilling mud) must be circulated in the hole. The kelly and drill pipe are hollow to allow fluid mixtures of clay and water, air, foam, and so forth, to be pumped through them to the bit. Drilling mud is stored in steel mud pits beside the rig. The drilling mud is then pumped down through the hollow drill pipes. It then carries the material to the surface through the space between the drill pipe and walls of the well (annulus). The swivel is a means to connect the mud hose to the drill string. *Casing* is a large-diameter concrete pipe that is inserted in the drill hole and prevents the hole from collapsing, and allows drilling mud to circulate. The bottom end of the drill string, *i.e.* the drill bit and drill collars, are also called *BHA* (Bottom Hole Assembly). The BHA may also contain other components such as a down-hole motor, and Measurement While Drilling (MWD) and Logging While Drilling (LWD) tools.

2.2 Drilling Rig

The drilling rig plays an important role in the drilling-well station. Drilling rigs may have a very complex structure varying in form and size. From the modeling point of view the rig imposes the boundary conditions on the drill string structure. A first approach to modeling the rig is to consider its geometric structure and the elements which constitute the rig. This may give an exact model, but it would not be practical. Because in fact this model would be very complex and numerically would be so slow that it could not be used in real-time applications. Another problem with this model is the fact that it cannot be used for another rig.

In [12] the model of two mass-spring-damper systems has been proposed for the model of longitudinal motions of the drilling rig, as shown in Figure 13.

Although the model is simple, it can provide a very good low frequency response (up to 20 Hz) which is quite enough for our purpose. Furthermore, when the rig changes, unlike the first method which needs a complete new model, here we just need a new identification for parameters of the model.

For the schematic diagram of Figure 13, the following model can be used.

$$\frac{dx_1}{dt} = u_1 - V_{in}$$

$$\frac{dx_2}{dt} = V_{tab} - u_1$$

$$m_1 \frac{du_1}{dt} = -k_1(x_1 + X_{in}) + k_2 x_2 - f_1 u_1 - m_1 g$$

$$m_2 \frac{dV_{tab}}{dt} = -k_2 x_2 + F_{top} - f_2 V_{tab} - m_2 g$$

where V_{in} and X_{in} are input variables representing initial longitudinal speed and initial displacement of the kelly; m_1, m_2, k_1, k_2, f_1, f_2 are parameters which should be identified for a given rig. g is the gravity acceleration. According to the experience gained during a few experiments, the masses m_1 and m_2 can be interpreted as the mass of the hook and the kelly, respectively [15]. F_{top} is the necessary force on the well surface to bore the drill string down into the borehole. V_{tab} is the kelly's longitudinal velocity.

Another important element of the rig is the rotary table which is modeled as a rotating mass with inertial momentum, as follows:

$$J_{tab} \frac{d\Omega_{tab}}{dt} = T_{in} - T_{top} - \kappa_{tab}\Omega_{tab}$$

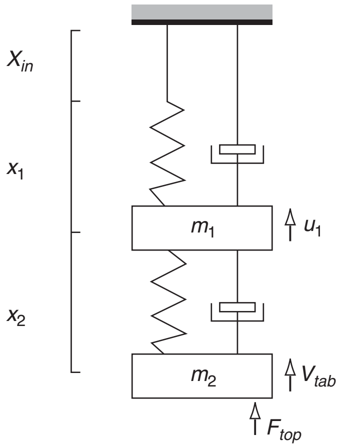

Figure 13

A mechanical model for the drilling rig.

where Ω_{tab} is the angular velocity of the rotary table and T_{in} is the torque applied on the rotary table. T_{top} is the torque needed to turn the drill string. k_{top} represents the coefficient of the Coulomb frictional resistance.

2.3 Drill Pipe

The drilling pipe is composed of multiple segments which are screwed together to construct a pipe of thousands of meters. Due its length, the drilling pipe exhibits torsion, longitudinal and lateral motions. In this paper, only longitudinal and torsion motions are considered. Precise modeling of the drilling pipe needs complicated methods such as finite element methods. In order to simplify the model, discretization of the drilling pipe to N = 15 sections is done: see Figure 14. This modeling approach fulfills the precision requirements with a minimum number of variables.

The following differential equations can be written to model the ith section, where $i \in [1, N-1]$.

$$M_i \frac{dV_i}{dt} = F_{i+1} - F_i - k_v V_i - M_i g$$

$$J_i \frac{d\Omega_i}{dt} = T_i - T_{i+1} - T_i^r$$

$$\frac{dF_i}{dt} = k_a(V_i - V_{i-1})$$

$$\frac{dT_i}{dt} = k_r(\Omega_{i-1} - \Omega_i)$$

where T_r^i is the friction-induced torque on the ith section and k_v, k_a and k_r are constants that depend on the physical characteristics of the pipe. M_i and J_i are the mass and the inertial momentum of the ith section of the pipe, respectively.

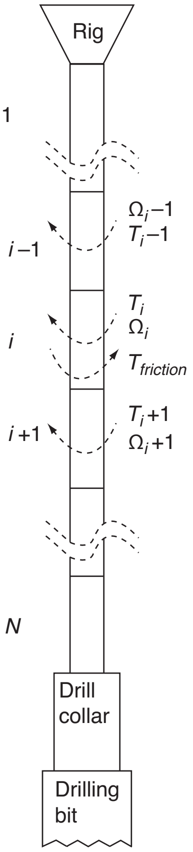

Figure 14

Discretization of the drilling pipe.

For the first section, *i.e.*, $i = 1$, we have:

$$M_1 \frac{dV_1}{dt} = F_2 - F_1 - k_v V_1 - M_1 g$$

$$J_1 \frac{d\Omega_1}{dt} = T_1 - T_2 - T_1^r$$

$$\frac{dF_1}{dt} = k_a (V_1 - V_{tab})$$

$$\frac{dT_1}{dt} = k_r (\Omega_{tab} - \Omega_1)$$

Note that $F_1 = F_{top}$ and $T_1 = T_{top}$. For the last section, *i.e.*, $i = N$, we have:

$$M_N \frac{dV_N}{dt} = F_c - F_N - k_v V_N - M_N g$$

$$J_N \frac{d\Omega_N}{dt} = T_N - T_c - T_N^r$$

$$\frac{dF_N}{dt} = k_a (V_N - V_{N-1})$$

$$\frac{dT_N}{dt} = k_r (\Omega_{N-1} - \Omega_N)$$

where F_c and T_c are the longitudinal force and angular torque that the collar applies on the lower end of the drill pipe, respectively.

The friction-induced torque on the drilling pipe sections T_r^i is given by the following empirical equations, for $i \in [1, N]$.

$$T_i^r = k_e L_i \Omega_i + k_d \left| P_i + \frac{M_i}{2} \right| (1 - e^{-\sigma|\Omega_i|}) \operatorname{sgn}(\Omega_i)$$

$$P_i = M_{bit} + M_c + \sum_{j=i+1}^{N} M_j$$

where k_d, k_e and σ are constants depending on the characteristics of the pipe. L_i is the length of the ith discretization section. P_i represents the mass hanging on the ith section. M_c and M_{bit} represent the mass of the collar and equivalent mass of the drilling bit, respectively.

2.4 Drill Collars

The drill collars, considered as rigid rods, are modeled in the same way as the drill pipe. Since the length of the drill collar is smaller than the length of the drill pipe, we do not discretize the drill collars and we consider a single rigid rod. Supposing that T_{bit} and F_{bit} are the torque and the force the drilling bit applies on the collar and Ω_c and V_c are the angular and longitudinal speed of the collar, then we can write the following equations for drill collars.

$$\begin{aligned}
M_c \frac{dV_c}{dt} &= F_{bit} - F_c k_v' V_c - M_c g \\
J_c \frac{d\Omega_c}{dt} &= T_c - T_{bit} - T_c^r \\
\frac{dF_c}{dt} &= k_a' (V_c - V_{bot}) \\
\frac{dT_c}{dt} &= k_r' (\Omega_{bot} - \Omega_c) \\
T_c^r &= k_e L_c \Omega_c + k_d' \left| M_{bit} + \frac{M_c}{2} \right| (1 - e^{-\sigma|\Omega_c|}) \operatorname{sgn}(\Omega_c)
\end{aligned}$$

where k'_v, k'_d and k'_r are constants depending on the characteristics of the collar. L_c is the length of the drill collar.

2.5 Drill Bit

The model of the rig, drill pipe and drill collars is composed of two uncoupled dynamics: a longitudinal and a rotational dynamics. These two dynamics should be coupled in the drill bit model. Thus, besides the longitudinal and rotational dynamics in the drill bit, a coupling dynamics is necessary. The diagram in Figure 15 shows these dynamics.

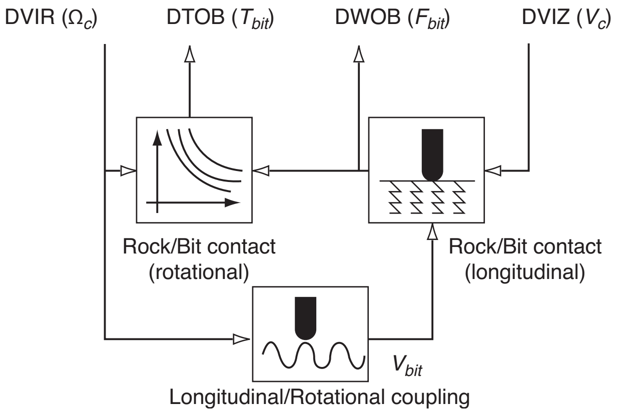
DVIR (Ω_c)
DTOB (T_{bit})
DWOB (F_{bit})
DVIZ (V_c)
Rock/Bit contact (rotational)
Rock/Bit contact (longitudinal)
V_{bit}
Longitudinal/Rotational coupling

Figure 15

Drilling bit model.

In Figure 15, *DVIZ*, *DVIR*, *DWOB* and *DTOB* represent the down-hole longitudinal velocity, the down-hole angular velocity, the down-hole weight-on-bit, and the down-hole torque-on-bit, respectively. The *DTOB*(T_{int}), which is the torque resistance against the rotation due to the rock/bit contact, is computed by iso-weight tables. These tables are used to compute the necessary torque as a function of *DWOB* and *DVIR*(Ω) Thus, we can write the following equation for T_{bit}.

$$T_{bit} = \text{Table\_Torque}(\Omega_c, DWOB)\,\text{sgn}(\Omega_c)(1 - e^{-\sigma|\Omega_c|})$$

The term $(1 - e^{-\sigma|\Omega_c|})$ has been introduced to have zero torque when the angular speed is zero.

The bottom end of the drilling bit is a tricone which transforms the rotational motions into longitudinal motions. Based on its geometric structure, we can write the following equations.

$$\theta = \int \Omega_c dt$$
$$V_{bit} = 3\Omega_c k_b \cos(3\theta)$$

where θ and V_{bit} are the angle of rotation of the bit and its speed in an axial direction, respectively. k_b is the amplitude of motion of the tricone bit.

The *WOB* is computed as a function of the axial speed of the tricone bit and the longitudinal speed of the bit. When the tricone is not in contact with the rock, *DWOB* is set to zero. The following equations have been proposed in [15].

$$\Delta_{bit} = \alpha \int (V_{bit} - V_c)dt$$
$$DWOB = \begin{cases} \Delta_{bit} & \text{if} \quad \Delta_{bit} \geq 0 \\ 0 & \text{if} \quad \Delta_{bit} < 0 \end{cases}$$

where Δ_{bit} represents the axial position of the drill bit. α is a constant coefficient representing the stiffness of the rock.

The resistance force which is applied on the drill collars is computed as a function of the *DWOB* and the friction between mud and the drilling bit.

$$F_{\text{bit}} = DWOB - k_m V_c |V_c|$$

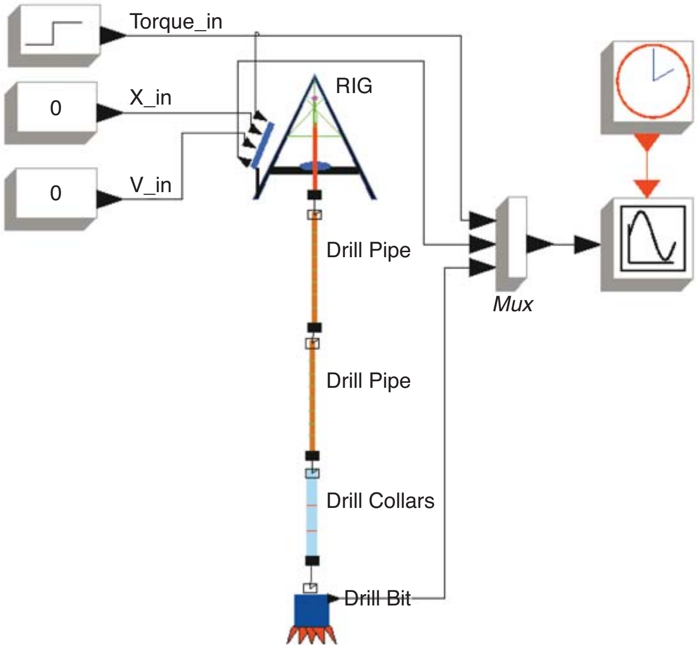

Figure 16

Model of the drilling rig in Scicos.

where k_m is the constant coefficient which depends on the mud characteristics. M_{bit} or the equivalent mass of the drilling bit is computed as a function of the WOB.

$$M_{\text{bit}} = -\frac{DWOB}{g}$$

3 APPLICATION

In order to construct the model of the drilling system, we need to develop four Modelica components, i.e., a rig block, a drill pipe block, a drill collar block, and a drill bit block. These blocks should be connected to build the model of the drill station. The Scicos diagram in Figure 16 shows the drilling station model in Scicos. In this model, there are four Modelica blocks and five causal (explicit) blocks. There are three blocks providing input variables of the rig block, i.e., the rotary table applied torque (Tin), the longitudinal speed of the kelly (Vin) and initial position of the kelly (Xin). There is also a scope block to visualize output variables of the model, and a clock block to activate the scope block to sample its inputs. Note that the connection between the implicit blocks are different from that of the explicit blocks. These connections represent physical connection.

The model, which is nonlinear and one-dimensional, can be easily adapted and used for the simulation of different rotary drilling cases. This model provides several bottom/surface transfer functions which can be used for real-time estimation of borehole variables. Furthermore, the model can be used for stability analysis, which is extremely important in controlling the drilling process. Another important use of the model is simulation of transient and steady-state behavior.

Figure 17 shows an unwanted phenomenon in rotary drilling, the result of torque fluctuations due to Coulomb friction application. This phenomenon is known as a stick-slip. The authors in [5, 15] show that this is in fact an oscillation or an instability in the drilling system. According to [5], the reason lies in the nonlinear friction between the bit and the rock.

The developed model provides a good framework for testing and simulation of oscillation phenomena in rotary drilling. In order to demonstrate these phenomena, a simulation is started at a steady-state angular speed of 66 rpm. The inputs are set to T_{in} = 3000 N.m, X_{in} = –0.03 m and V_{in} = 0 m/s. With these inputs, the system is stable. At time = 100 s, the torque is increased to 3200 N.m. which sets off an oscillation in the rotational speed at the bottom hole as well as at the well surface. At time = 200 s, although the input torque is reduced back to T_{in} = 3000 N.m, the drill pipe remains in the oscillatory mode. Note that the oscillation amplitude at the bottom hole is much larger than what is seen at the surface, which shows the importance of prediction and detection of oscillations before the system failure.

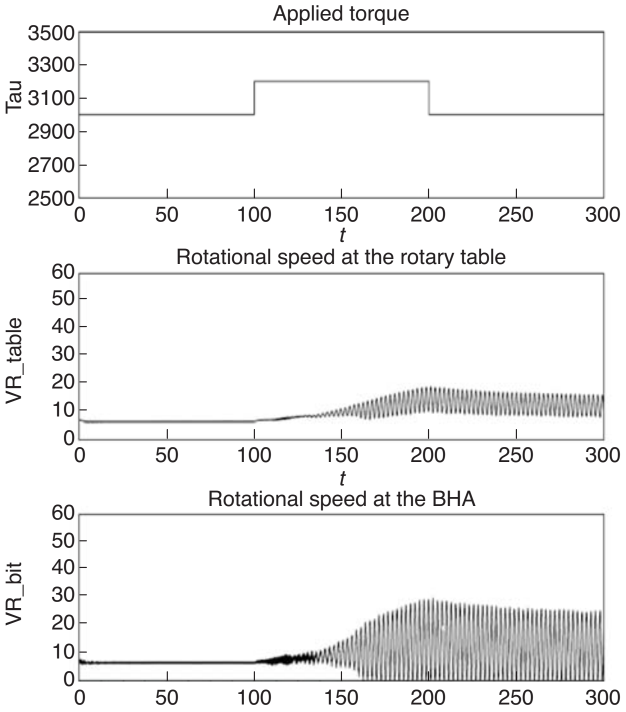

Figure 17

Simulation of a drilling showing the oscillation in the rotation speed.

CONCLUSION

In this paper, we show with two industrial examples that modeling in Modelica has several advantages: it is a declarative language with which a very general hybrid system can be modeled. The Modelica models are independent of the simulation tool and can be simulated in any Modelica simulator. In the Scicos simulator, Modelica models are constructed with implicit blocks. The overall model is then built by interconnecting the implicit blocks as an engineer does.

Modeling using the acausal approach (component-based) has the advantage that there are similarities between the model and the physical system. Unlike other modeling anguages, where a small change in the physical system may lead to big changes in the model, in Modelica, writing quations in the implicit form is easy to implement, and to modify.

Another important advantage of Modelica lies in the symbolic manipulation of models. The reason lies in giving several possibilities such as index reduction and generation of the analytical Jacobian, which speeds up the simulation.

ACKNOWLEDGEMENTS

This work was carried out in the RNTL SimPA2-C6E2 project, supported by ANVAR and the Scientific Management of IFP. SIMPA2–C6E2 is the continuation of the RNTL project SIMPA, a platform constructed with the help of the Free Open-source software Scilab, Scicos and Modelica.

The engine model was proposed by Gilles Corde (Director of the Signal Processing, Automation and Control Department in IFP). The model was first modeled by the Scicos standard hybrid and then in 2007 by the new Modelica/Scicos version.

The authors gratefully acknowledge the help of Isabelle Rey-Fabret (research professor at IFP/École du pétrole et des moteurs) in understanding the drilling-well model.

REFERENCES

1 Aquino C. (1981) *Transient A/F control characteristics of the 5 liter central fuel injection engine*, SAE Technical Paper Series.

2 Benjelloun-Dabaghi Z., Lacabanne B. (2003) *Projet RNTL-SIMPA : Validation d'une nouvelle plateforme de simulation de systèmes hybrides implicites par des cas industriels*, Rapport IFP 57704.

3 Brenan K.E., Campbell S.L., Petzold L.R. (1996) *Numerical solution of Initial – Value Problems in Differential – Albebric Equations,* Classics in Applied Mathematics, Siam eds.

4 Campbell S.L., Chancelier J.P., Nikoukhah R. (2005) *Modeling and simulation Scilab/Scicos*, Springer Verlag, 1st ed.

5 Dawson R., Lin Y.Q., Spanos P. (1987) *Drill string oscillation*.

6 Dupraz P. (1998) Modélisation et commande avancées d'un moteur diesel à injection directe, *Thèse de l'Institut National Polytechnique de Grenoble*.

7 Gissinger G., Le Fort-Piat N. (2002) *Contrôle-commande de la voiture, Traité IC2, série Systèmes automatisés*, Hermès Sciences, Paris.

8 Fritzson P. (2004) *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, Wiley-IEEE Press, 1st ed., ISBN 0-471-47163-1.

9 Heywood J.B. (1988) *Internal Combustion Engine Fundamentals*, McGraw-Hill series in mechanical engineering, international edition.

10 Jerrelind J. (1998) A Study in Model Based Control of an Evaporative Emissions Management System, *Master's thesis*, Department of Electrical Engineering, Linköping University.

11 Najafi M. (2005) The Numerical Solver for the Simulation of the Hybrid Dynamical Systems, *Doctor of science thesis*, Paris XII University.

12 Nakayama Y., Boucher F. (1999) *Introduction to Fluid Mechanics*, Arnold, London, Great Britain.

13 Nguyen J.-P. (1996) *Drilling*. *"Oil and Gas Field Development Techniques"*, ed. Technip.

14 Silverlind D. (2001) Mean Value Engine Modeling with Modelica, *Master's thesis*, Department of Electrical Engineering, Linkoping University.

15 Rey-Fabret I., Mabile C., Oudin N. (1997) Detecting whirling behaviour of the drill string from surface measurements, *SPE-72nd annual technical conference and exhibition of the society of Petroleum Engineers*, San Antonio, Oct. 5-8 1997, Proceedings, pp. 223-231.

Final manuscript received in May 2008
Published online in October 2008